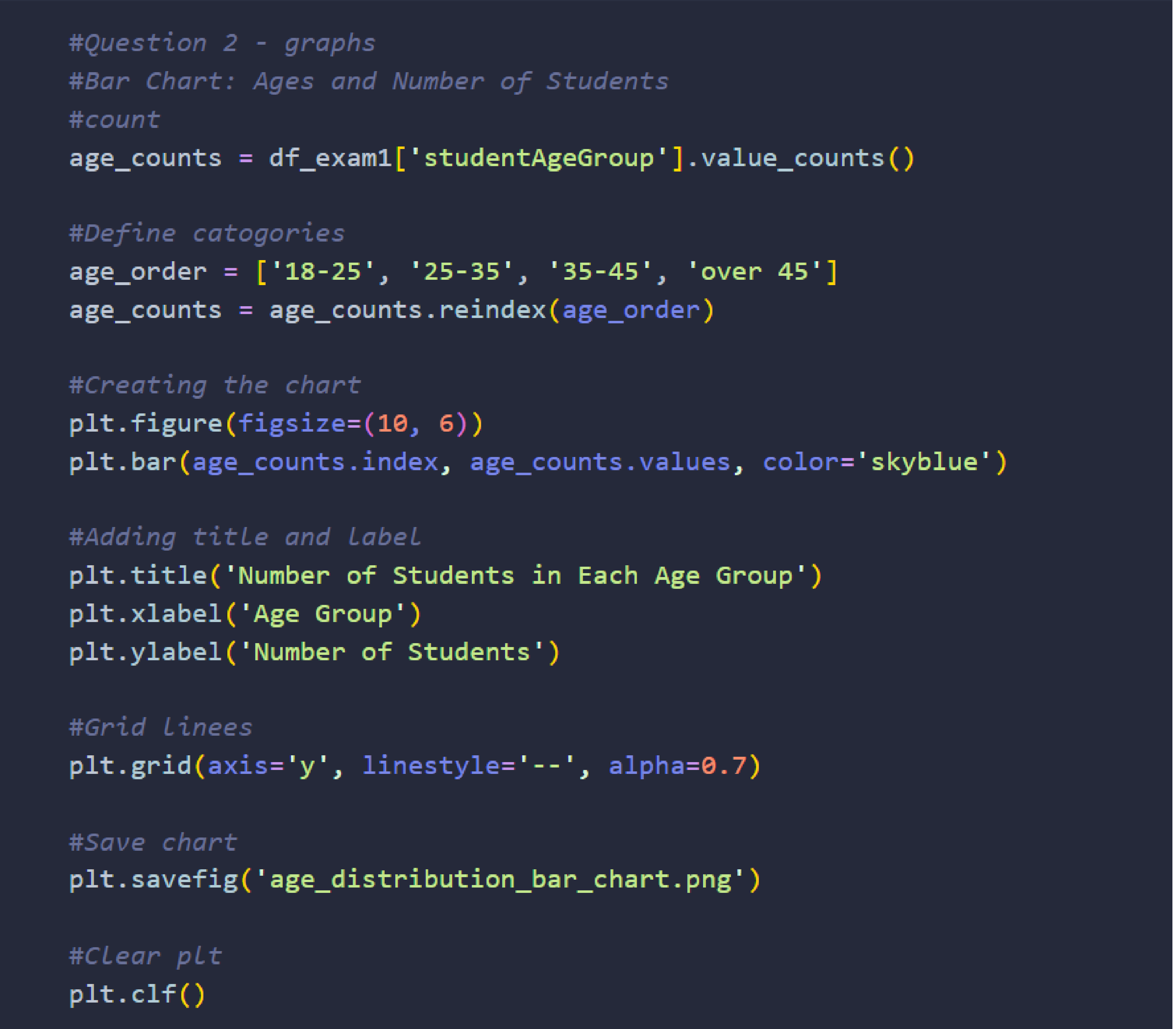
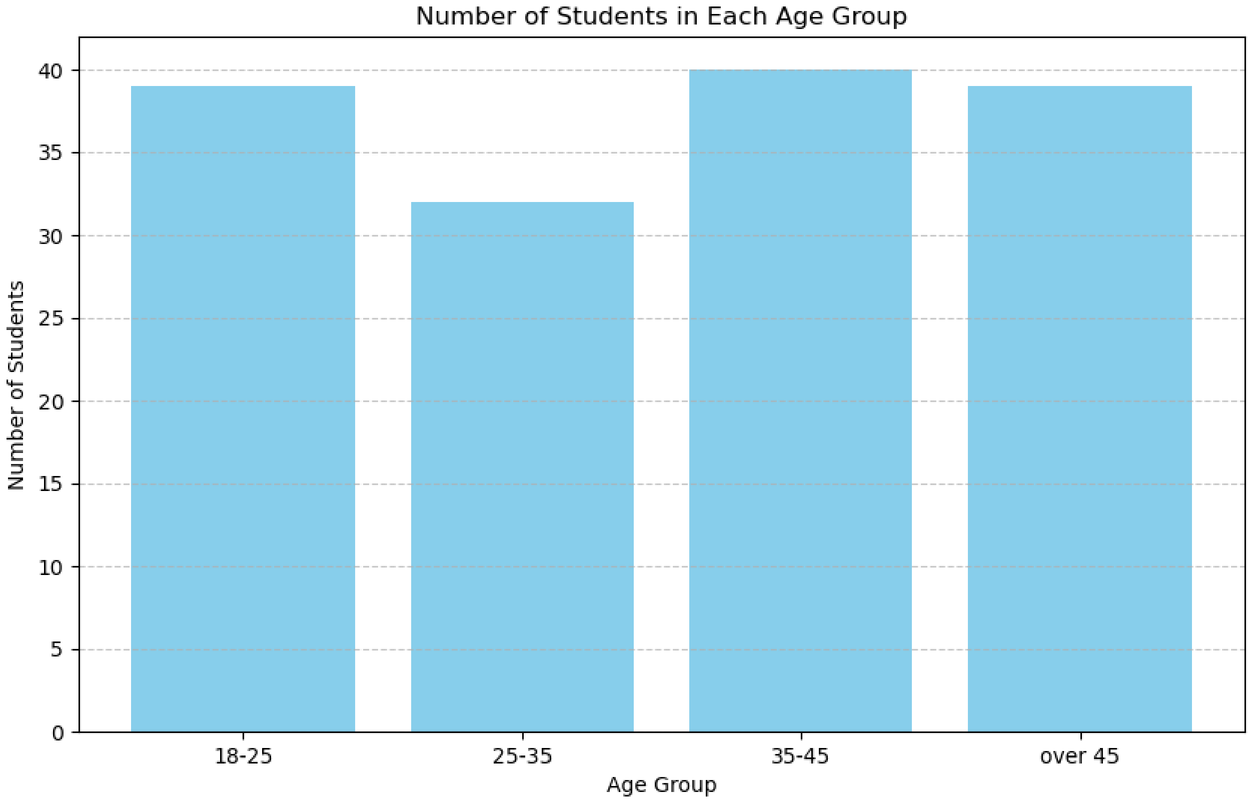
Bar Chart code snippet

Note: The data in these tables is completely random, so it can't be used to find any real-world insights. The only thing you could study from it is how well (or poorly) the computer actually generates random numbers.

This code snippet shows the cell where the bar chart (next page) is created and saved as a PNG to the root of the project folder.

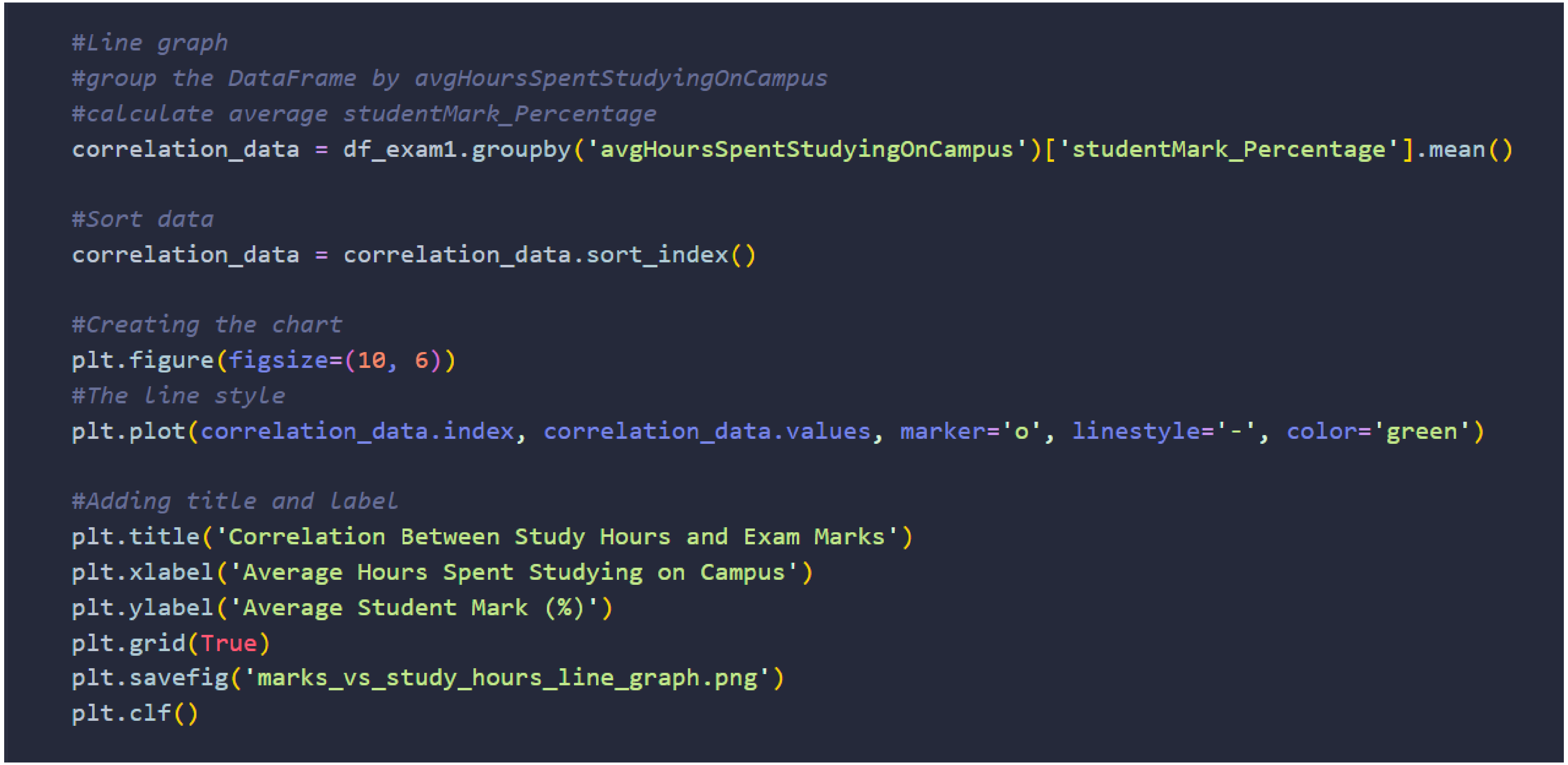


```
#Question 2 - graphs
#Bar Chart: Ages and Number of Students
#count
age_counts = df_exam1['studentAgeGroup'].value_counts()

#Define catogories
age_order = ['18-25', '25-35', '35-45', 'over 45']
age_counts = age_counts.reindex(age_order)

#Creating the chart
plt.figure(figsize=(10, 6))
plt.bar(age_counts.index, age_counts.values, color='skyblue')

#Adding title and label
plt.title('Number of Students in Each Age Group')
plt.xlabel('Age Group')
plt.ylabel('Number of Students')

#Grid linees
plt.grid(axis='y', linestyle='--', alpha=0.7)

#Save chart
plt.savefig('age_distribution_bar_chart.png')

#Clear plt
plt.clf()
```



Plot analysis

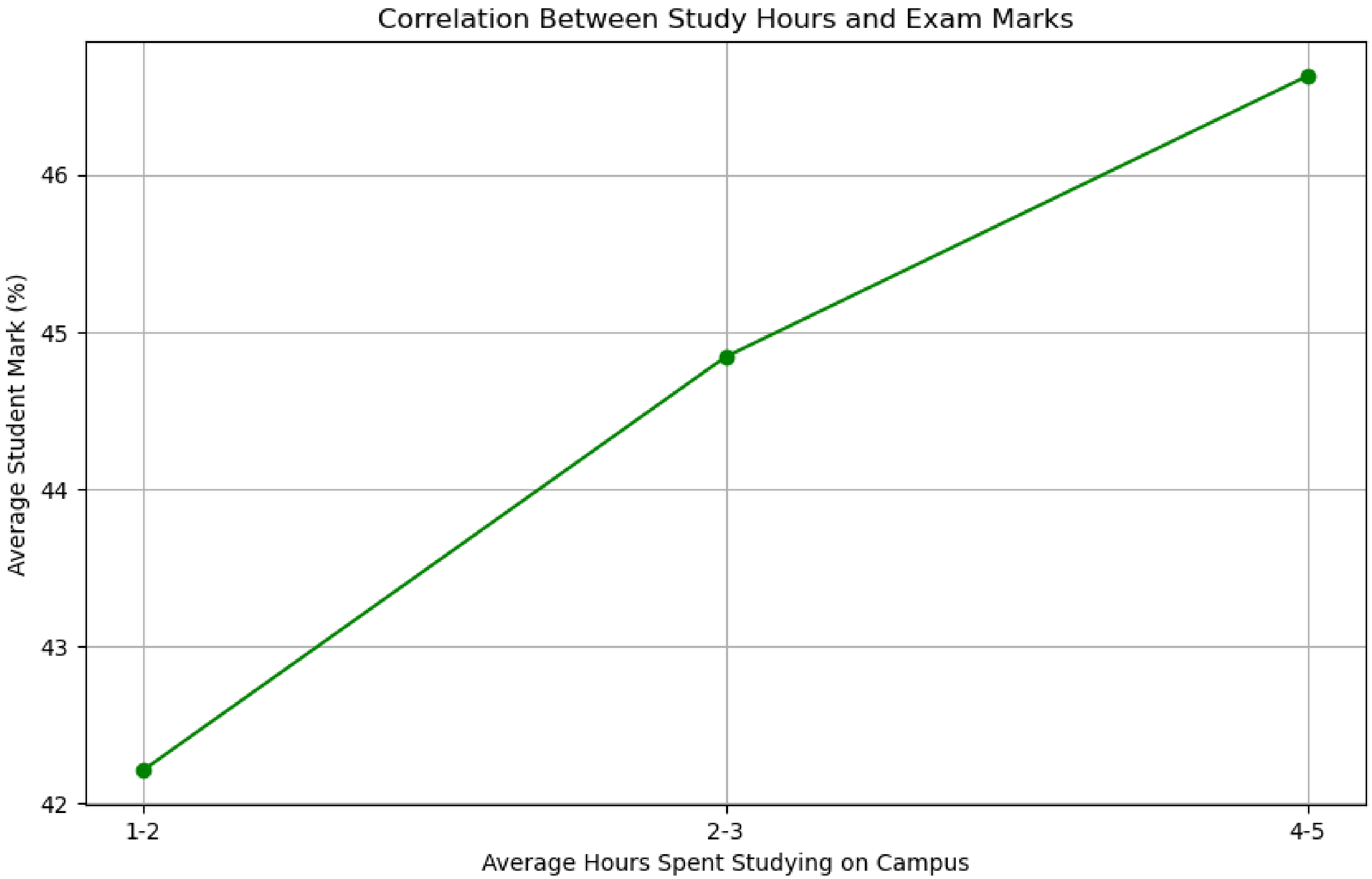
A couple of useful observations can be made about the student age demographic shown in the bar chart: The 35–45 age group has the highest number of students, while the 25–35 age group has the fewest. The remaining two groups, 18–25 and 45+, have nearly identical, lower student counts.



```
#Line graph
#group the DataFrame by avgHoursSpentStudyingOnCampus
#calculate average studentMark_Percentage
correlation_data = df_exam1.groupby('avgHoursSpentStudyingOnCampus')['studentMark_Percentage'].mean()

#Sort data
correlation_data = correlation_data.sort_index()

#Creating the chart
plt.figure(figsize=(10, 6))
#The line style
plt.plot(correlation_data.index, correlation_data.values, marker='o', linestyle='-', color='green')

#Adding title and label
plt.title('Correlation Between Study Hours and Exam Marks')
plt.xlabel('Average Hours Spent Studying on Campus')
plt.ylabel('Average Student Mark (%)')
plt.grid(True)
plt.savefig('marks_vs_study_hours_line_graph.png')
plt.clf()
```

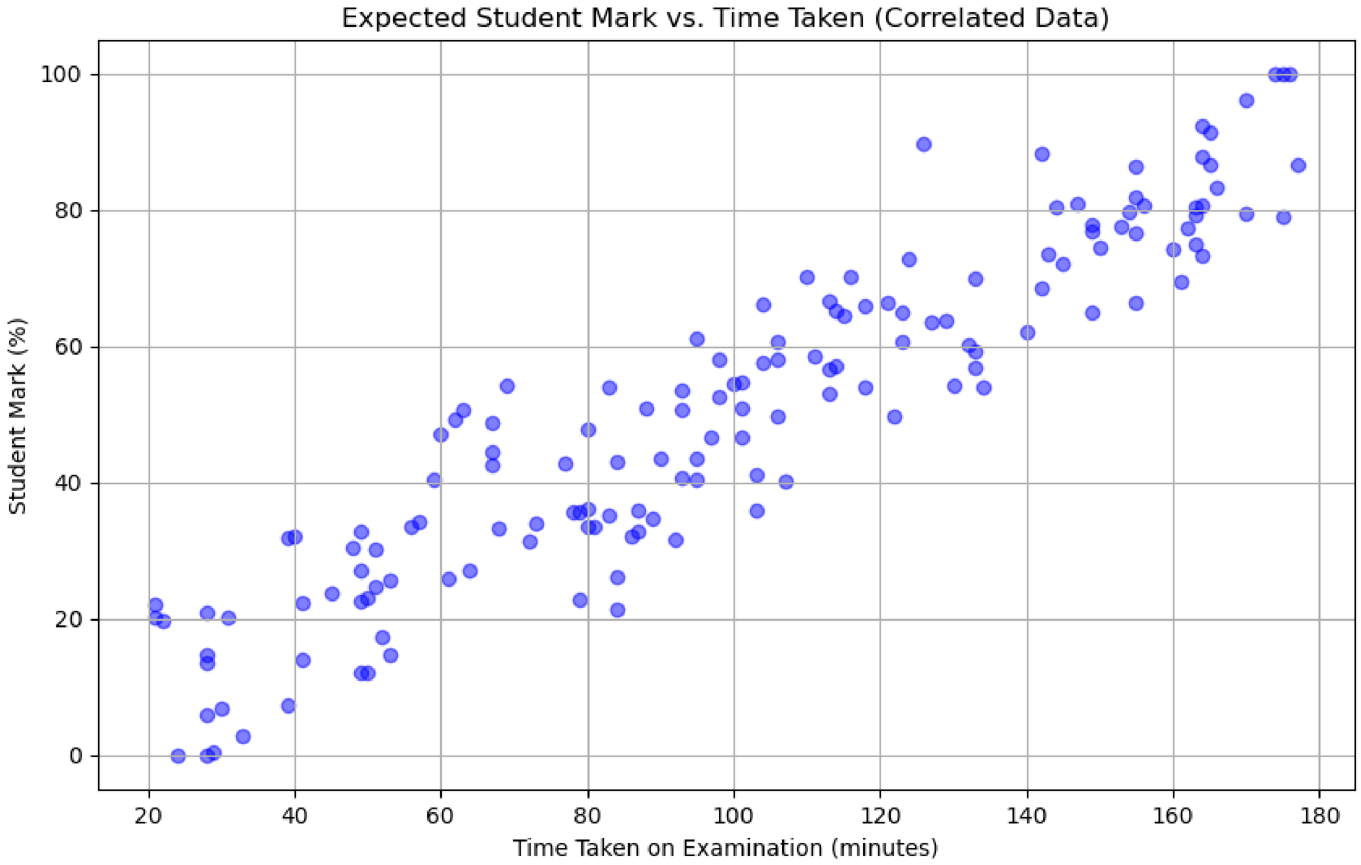
Line Chart code snippet

This code snippet shows the cell where the bar chart (next page) is created and saved as a PNG to the root of the project folder.



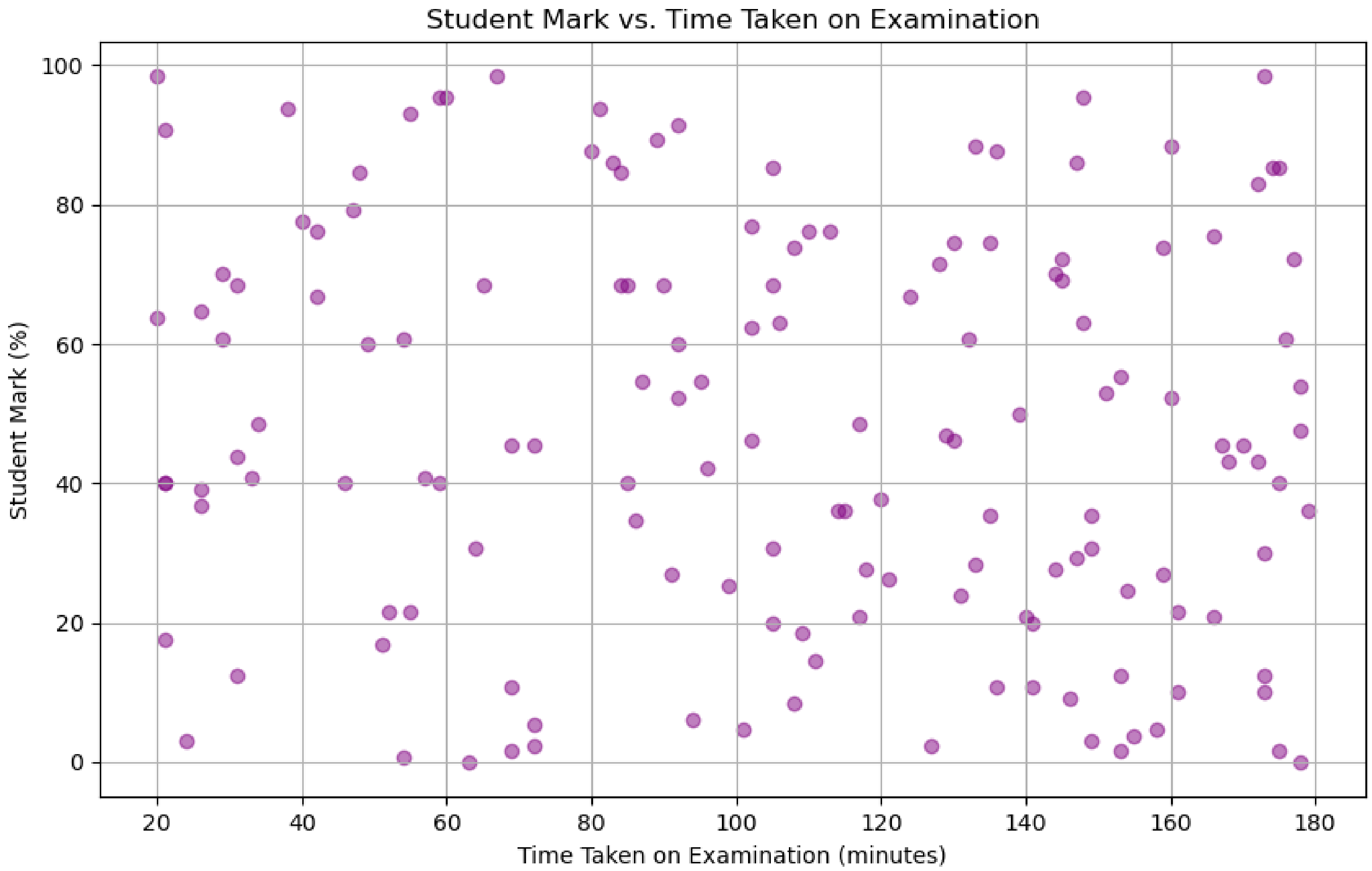
Plot analysis

This line graph shows the correlation between study hours, and Exam marks. We can observe a slight positive correlation between more hours studying and the students exam mark. We can see that the 1-2 hours of studying has the lowest mark, which is what we would expect with real world data. The biggest improvement that can be observed in this line graph is from 1-2 hours to 2-3 hours of studying, where the students marks raise by approximately three percent.



Example Scatter Chart

Since the data in my project was randomly generated, this scatter plot does not show any useful information. I decided to fabricate one what demonstrates the correlation I would expect from real world data. It can be observed in this example chart that there is a very strong correlation between the time taken on the exam, and the student's mark. On the next page is the scatter chart from my project



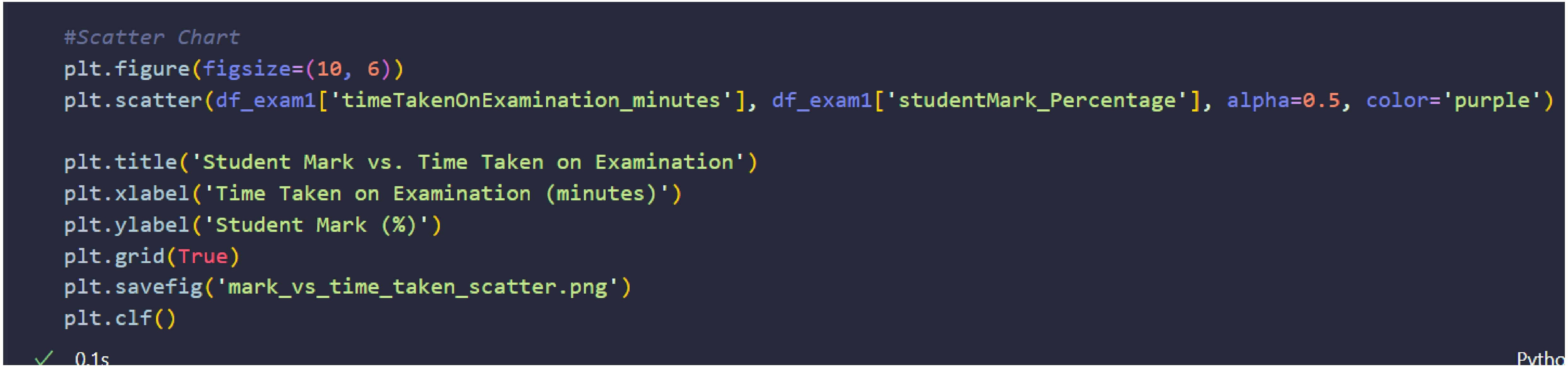
Scatter Plot analysis

As I explained on the previous page, we can see that there is no observable correlation between Time taken on the exam, and the students exam mark. This is a result of the data being generated randomly.

Students who finished quickly (20-40 minutes) achieved marks ranging from very low (under 10%) to very high (over 80%). The same is true for students who took the longest (160-180 minutes).

Scatter Chart code snippet

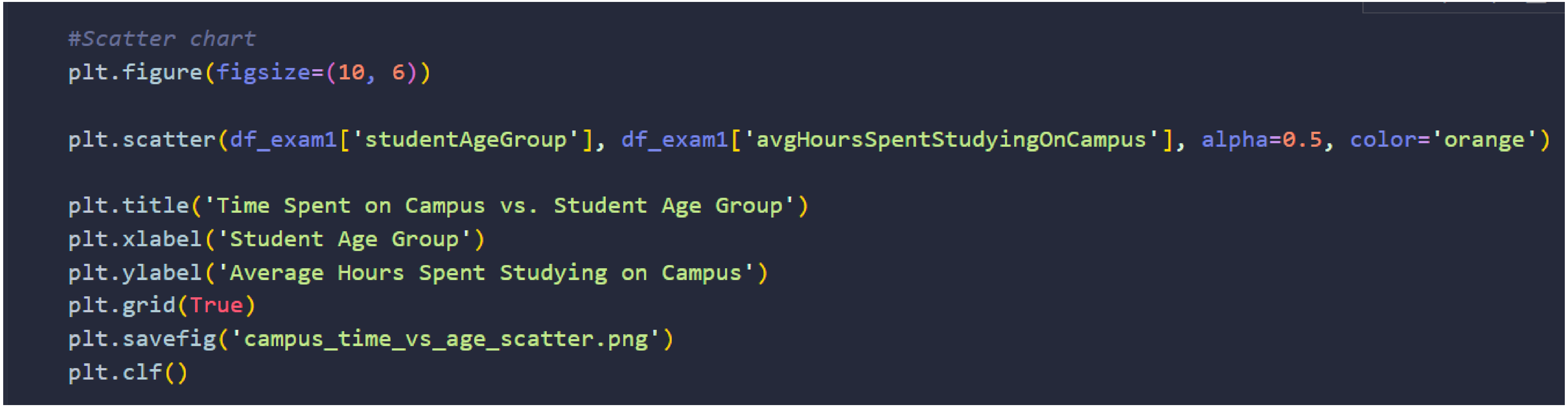
This code snippet shows the cell where the scatter chart (previous page) is created and saved as a PNG to the root of the project folder.



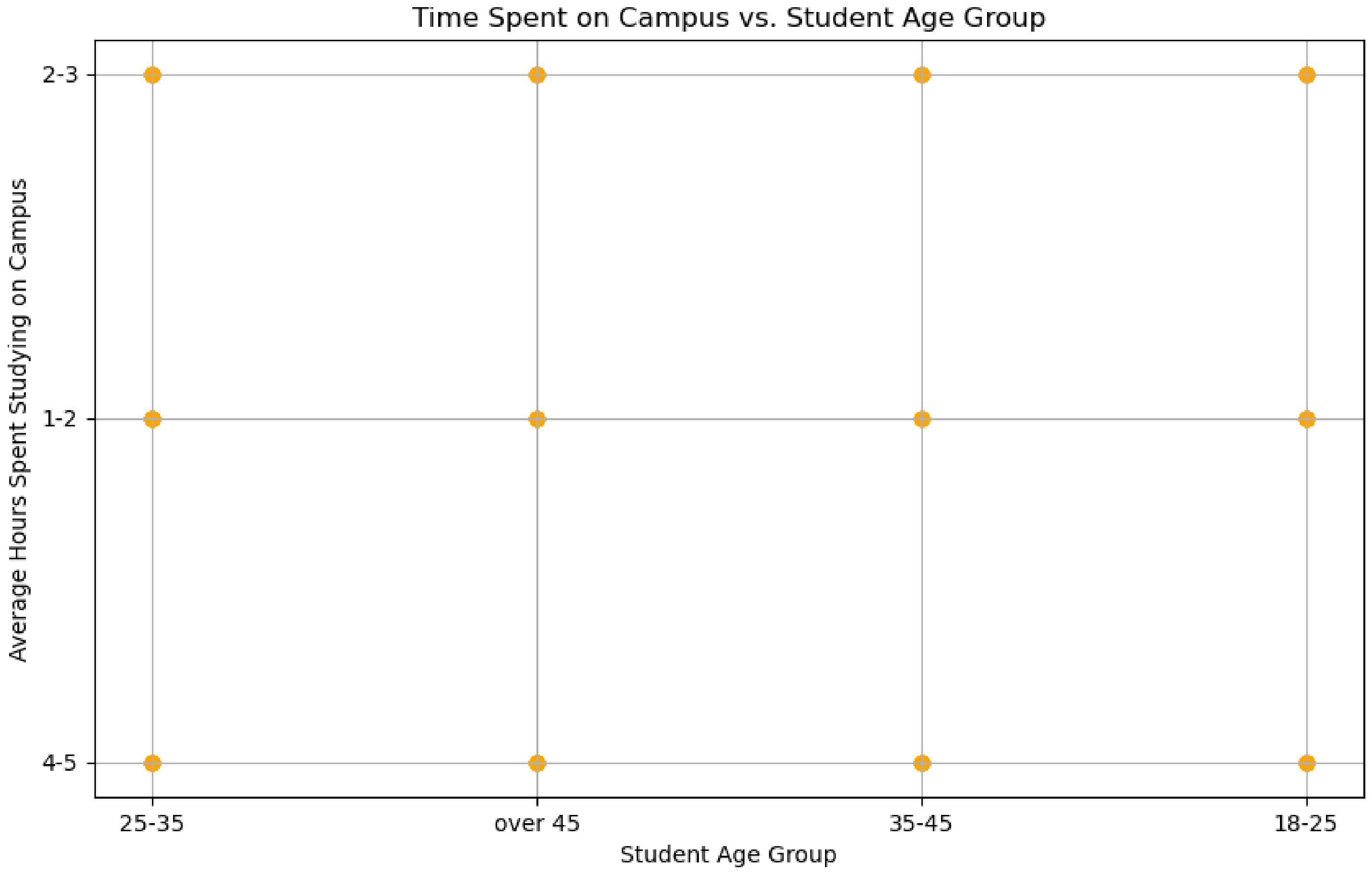
```
#Scatter Chart
plt.figure(figsize=(10, 6))
plt.scatter(df_exam1['timeTakenOnExamination_minutes'], df_exam1['studentMark_Percentage'], alpha=0.5, color='purple')

plt.title('Student Mark vs. Time Taken on Examination')
plt.xlabel('Time Taken on Examination (minutes)')
plt.ylabel('Student Mark (%)')
plt.grid(True)
plt.savefig('mark_vs_time_taken_scatter.png')
plt.clf()
```

Scatter Chart code snippet

This code snippet shows the cell where the scatter chart (next page) is created and saved as a PNG to the root of the project folder.



```
#Scatter chart
plt.figure(figsize=(10, 6))

plt.scatter(df_exam1['studentAgeGroup'], df_exam1['avgHoursSpentStudyingOnCampus'], alpha=0.5, color='orange')

plt.title('Time Spent on Campus vs. Student Age Group')
plt.xlabel('Student Age Group')
plt.ylabel('Average Hours Spent Studying on Campus')
plt.grid(True)
plt.savefig('campus_time_vs_age_scatter.png')
plt.clf()
```



Scatter Plot analysis

The graph just confirms that all 12 possible combinations (4 age groups x 3 study hour categories) are present in the 150 student sample.

I expected this result since I saw the question as there are only 4 age groups and 3 study hour categories, all 150 student data points are plotted at exactly the same 12 (x, y) coordinates. Once again there are no useful observations can be made from this plot.